

IN THE CLAIMS

Claims 1-38 are presented below, with claims 5-6 and 15-16 pending. As shown below, claims 5 and 15 have been amended, claims 1-4, 7-14, and 17-38 have been canceled.

1. – 4. (Canceled)

5. (Currently Amended) A data processing apparatus according to Claim 4, further comprising for executing an object-oriented operating system made up of a plurality of objects among which messages are communicated, said data processing apparatus comprising:

means for rendering an object, which has received a combining request message requesting addition of a predetermined object, to create a table data structure used for referencing to said predetermined object as a component object, and to initialize said table data structure with data of said component object, thereby constituting a composite object;

means for creating a data structure of at least one said component object, registering the data structure in said table data structure, and registering a relationship between at least one message processing function possessed by said component object and a message interface for requesting the message processing function in the data structure of said component object;

means for checking, when a predetermined object is added to said composite object, whether a sequential execution relation exists between said predetermined object and all of component objects making up said composite object; and

means for adding said predetermined object after the sequential execution relation has been confirmed such that said predetermined object is not added to said composite object if said

sequential execution relation does not exist between said predetermined object and all of component objects making up said composite object;

wherein said composite object has a thread and each component object of said composite object uses that thread for communicating with one another;

wherein said composite object has a specific execution thread and executes message processing issued to said component object with said specific execution thread; and

wherein upon receiving a request for adding a predetermined component object, said composite object additionally registers a data structure of said predetermined component object in said table data structure.

6. (Original) A data processing apparatus according to Claim 5, wherein the sequential execution relation is checked by means for checking the fact that at the time when a message is transmitted to said predetermined object, all of the component objects making up said composite object are not required to run in parallel to said predetermined object, means for checking the fact that at the time when said predetermined object transmits a message to one of the component objects making up said composite object, the one component object is never already under processing of another message, and means for checking the fact that at the time when said predetermined object receives a message from one of the component objects making up said composite object, said predetermined object is never already under processing of another message.

7. – 14. (Canceled)

15. (Currently Amended) A data processing method ~~according to Claim 14, further executes a~~
~~step of~~ for an object-oriented operating system made up of a plurality of objects among which
messages are communicated, said data processing method comprising the steps of:

rendering an object, which has received a combining request message requesting addition
of a predetermined object, to create a table data structure used for referencing to said
predetermined object as a component object, and to initialize said table data structure with data
of said component object, thereby constituting a composite object;

creating a data structure of at least one said component object, registering the data
structure in said table data structure, and registering a relationship between at least one message
processing function possessed by said component object and a message interface for requesting
the message processing function in the data structure of said component object;

checking, when a predetermined object is added to said composite object, whether a
sequential execution relation exists between said predetermined object and all of component
objects making up said composite object; ~~and a step of~~

adding said predetermined object after the sequential execution relation has been
confirmed such that said predetermined object is not added to said composite object if said
sequential execution relation does not exist between said predetermined object and all of
component objects making up said composite object;

wherein said composite object has a thread and each component object of said composite
object uses that thread for communicating with one another;

wherein said composite object has a specific execution thread and executes message
processing issued to said component object with said specific execution thread; and

wherein upon receiving a request for adding a predetermined component object, said composite object additionally registers a data structure of said predetermined component object in said table data structure.

16. (Original) A data processing method according to Claim 15, wherein the sequential execution relation is checked by a step of checking the fact that at the time when a message is transmitted to said predetermined object, all of the component objects making up said composite object are not required to run in parallel to said predetermined object, a step of checking the fact that at the time when said predetermined object transmits a message to one of the component objects making up said composite object, the one component object is never already under processing of another message, and a step of checking the fact that at the time when said predetermined object receives a message from one of the component objects making up said composite object, said predetermined object is never already under processing of another message.

17. – 38. (Canceled)